Multi-Mode Tensor Representation of Motion Data

Björn Krüger*, Jochen Tautges*, Meinard Müller† and Andreas Weber*

*Institut für Informatik, Universität Bonn
Römerstr. 164, 53117 Bonn, Germany
email: {kruegerb | tautges | weber@cs.uni-bonn.de
www:http://cg.cs.uni-bonn.de

†Max-Planck-Institut Informatik
Campus E1-4, 66123 Saarbrücken, Germany
www: http://www.mpi-inf.mpg.de
meinard@mpi-inf.mpg.de

Abstract

In this paper, we investigate how a multilinear model can be used to represent human motion data. Based on technical modes (referring to degrees of freedom and number of frames) and natural modes that typically appear in the context of a motion capture session (referring to actor, style, and repetition), the motion data is encoded in form of a high-order tensor. This tensor is then reduced by using N-mode singular value decomposition. Our experiments show that the reduced model approximates the original motion better then previously introduced PCA-based approaches. Furthermore, we discuss how the tensor representation may be used as a valuable tool for the synthesis of new motions.

Keywords: Motion Capture, Tensor Representation, N-Mode SVD, Motion Synthesis

First presented at the 2nd International Conference on Computer Graphics Theory and Applications (GRAPP 2007) extended and revised for JVRB.

1 Introduction

Motion capture or mocap systems allow for tracking and recording of human motions at high spatial and temporal resolutions. The resulting 3D mocap data is used for motion analysis in fields such as sports sciences, biomechanics, or computer vision, and in particular for motion synthesis in data-driven computer animation. In the last few years, various morphing and blending techniques have been suggested to modify prerecorded motion sequences in order to create new, naturally looking motions, see, e.g., [GP00, Tro02, KGP02, SHP04, KG04, OBHK05, MZF06, CH07, SH07].

In view of motion reuse in synthesis applications, questions concerning data representation, data organization, and data reduction as well as content-based motion analysis and retrieval have become important topics in computer animation. In this context, motion representations based on *linear models* as well as dimensionality reduction techniques via principal component analysis (PCA) have become well-established methods [BSP+04, CH05, FF05, LZWM05, SHP04, GBT04, Tro02, OBHK05]. Using these linear methods one neglects information of the motions sequences, such as the temporal order of the frames or information about different actors whose motions are included within the database.

In the context of facial animation, Vlasic et al. [VBPP05] have successfully applied *multilinear models* of 3D face meshes that separably parameterizes semantic aspects such as identity, expression, and visemes. The strength of this technique is that additional information can be kept within a multilinear model. For example, classes of semantically related motions can be organized by means of certain modes that naturally correspond to semantic aspects referring to an actor's identity or a particular motion style. Even though multilinear models are a suitable tool for incorporating such aspects into a unified framework, so far only little work has been done to employ these techniques for motion data [Vas02, RCO05, MK06].

In this paper, we introduce a multi-linear approach for modeling classes of human motion data. Encoding the motion data as a high-order tensor, we explicitly account for the various modes (e. g., actor, style, repetition) that typically appear in the context of a motion capture session. Using standard reduction techniques based on multi-mode singular value decomposition (SVD), we show that the reduced model approximates the original motion better then previously used PCA-reduced models. Furthermore, we sketch some applications to motion synthesis to demonstrate the usefulness of the multilinear model in the motion context.

The idea of a tensor is to represent an entire class of semantically related motions within a unified framework. Before building up a tensor, one first has to establish temporal correspondence between the various motions while bringing them to the same length. This task can be accomplished by techniques based on dynamic time warping [BW95, GP00, KG03, HPP05]. Most features used in this context are based on spatial or angular coordinates, which are sensitive to data variations that may occur within a motion class. Furthermore, local distance measures such as the 3D point cloud distance as suggested by Kovar and Gleicher [KG03] are computationally expensive. In our approach, we suggest a multiscale warping procedure based on physics-based motion parameters such as center of mass acceleration and angular momentum. These features have a natural interpretation, they are invariant under global transforms, and show a high degree or robustness to spatial motion variation. As a further advantage, physics-based features are still semantically meaningful even on a coarse temporal resolution. This fact allows us to employ a very efficient multiscale algorithm for the warping step. Despite of these advantages, only few works have considered the physics-based layer in the warping context, see [MZF06, SH05].

The remainder of this paper is organized as follows. In Section 2, we introduce the tensor-based motion representation and summarize the data reduction procedure based on singular value decomposition (SVD). The multiscale approach to motion warping using physics-based parameters is then described in Section 3. We have conducted experiments on systematically recorded motion capture data. As representative examples, we discuss three motion classes including walking, grabbing, and cartwheel motions, see Section 4. We conclude with Section 5, where we indicate future research directions. In particular, we discuss possible strategies for the automatic generation of suitable motion classes from a scattered set of motion data, which can then be used in our tensor representation.

2 Multilinear Algebra

Our tensor representation is based on multilinear algebra, which is a natural extension of linear algebra. A *tensor* Δ of *order* $N \in \mathbb{N}$ and type $(d_1, d_2, \ldots, d_N) \in \mathbb{N}^N$ over the real number $\mathbb{R}$ is defined to be an element in $\mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$. The number $d := d_1 \cdot d_2 \cdot \ldots \cdot d_N$ is referred to as the *total dimension* of Δ. Intuitively, the tensor Δ represents d real numbers in a multidimensional array based on N indices. These indices are also referred to as the *modes* of the tensor Δ. As an example, a vector $v \in \mathbb{R}^d$ is a tensor of order $N = 1$ having only one mode. Similarly, a matrix $M \in \mathbb{R}^{d_1 \times d_2}$ is a tensor of order $N = 2$ having two modes which correspond to the columns and rows. A more detailed description of multilinear algebra is given in [VBPP05].

2.1 Tensor construction

In our context, we deal with 3D human motion data as recorded by motion capture systems. A (sampled) motion sequence can be modelled as a matrix $M \in \mathbb{R}^{n \times f}$, where the integer $n \in \mathbb{N}$ refers to the degrees of freedom (DOFs) needed to represent a pose of an underlying skeleton (e. g. encoded by Euler angles or quaternions) and the integer $f \in \mathbb{N}$ refers to the number of frames (poses) of the motion sequence. In other words, the ith colum of M, in the following also denoted by $M(i)$, contains the DOFs of the ith pose, $1 \leq i \leq f$. In the following examples, we will work either with an Euler angle representation of a human pose having $n = 62$ DOFs or with a quaternion representation having $n = 119$ DOFs (with $n = 4 \cdot m + 3$ where $m = 29$

is the number of quaternions representing the various joint orientations). In both representations 3 DOFs are used to describe the global 3D position of the root node of the skeleton.

We now describe how to construct a tensor from a given class of semantically related motion sequences. After a warping step, as will be explained in Section 3, all motion sequences are assumed to have the same number of frames. We will introduce two types of modes referred to as *technical modes* and *natural modes*. We consider two technical modes that correspond to the degrees of freedom and number of frames, respectively:

- **Frame Mode**: This mode refers to the number of frames a motion sequence is composed of. The dimension of the Frame Mode is denoted by f.
- **DOF Mode**: This mode refers to the degrees of freedom, which depends on the respective representation of the motion data. The dimension of the DOF Mode is denoted by n.

Sometimes the two technical modes are combined to form a single mode, which is referred to as *data mode*:

- **Data Mode**: This mode refers to an entire motion sequence, where all motion parameters are stacked into a single vector. For a motion sequence $M \in \mathbb{R}^{n \times f}$, the dimension of the Data Mode is $f \cdot n$.

Additionally, we introduce natural modes that typically appear in the context of a motion capture session:

- **Actor Mode:** This mode corresponds to the different actors performing the motion sequences. The dimension of the actor mode (number of actors) is denoted by a.
- **Style Mode:** This mode corresponds to the different styles occurring in the considered motion class. The meaning of style differs for the various motion classes. The dimension of the style mode (number of styles) is denoted by s.
- **Repetition Mode:** This mode corresponds to the different repetitions or interpretations, which are available for a specific actor and a specific style. The dimension of the repetition mode (number of repetitions) is denoted by r.

The natural modes correspond to semantically meaningful aspects that refer to the entire motion sequence. These aspects are often given by some rough textual description or instruction. The meaning of the modes may depend on the respective motion class. Furthermore, depending on the availability of motion data and suitable metadata, the various modes may be combined or even further subdivided. For example, the style mode may refer to emotional aspects (e. g., furious walking, cheerful walking), motion speed (e. g., fast walking, slow walking), motion direction (e. g., walking straight, walking to the left, walking to the right), or other stylistic aspects (e. g., limping, tiptoeing, marching). Further example will be discussed in Section 4. Finally, we note that in [MK06] the authors focus on correlations with respect to joints and time only, which, in our terminology, refer to the technical modes. Furthermore, in [Vas02], the authors discuss only a restricted scenario considering leg movements in walking motions.

2.2 N-Mode SVD

In our experiments, we constructed several data tensors with different numbers of modes from the data base described in Section 4.1. The tensor with the smallest number of modes was created by using the three natural modes (Actors, Style, and Repetition) and the Data Mode. With this arrangement we obtain a tensor in the size of $f \cdot n \times a \times s \times r$. It is also possible to use the Frame and the DOF Mode, instead of the Data Mode, to arrange the same motions sequences within the tensor. The natural modes are not changed when using this strategy. Therefore a tensor of this type has a size of $f \times n \times a \times s \times r$.

Similar to [VBPP05], a data tensor Δ can be transformed by an N-mode singular value decomposition (N-mode SVD). Recall that Δ is an element in $\mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$. The result of the decomposition is a *core tensor* Φ of the same type and associated orthonormal matrices $U_1, U_2, \ldots, U_N$. The matrices U_k are elements in $\mathbb{R}^{d_k \times d_k}$ where $k \in \{1, 2, \ldots, N\}$. The tensor decomposition in our experiments was done by using the N-way Toolbox [BA00]. Mathematically this decomposition can be expressed in the following way:

$$\Delta = \Phi \times_1 U_1 \times_2 U_2 \times_3 \ldots \times_N U_N.$$

This product is defined recursively, where the mode-k-multiplication $\times_k$ with U_k replaces each mode-k-

vector v of $\Phi \times_1 U_1 \times_2 U_2 \times_3 \ldots \times_{k-1} U_{k-1}$ for $k > 1$ (and Φ for $k = 1$) by the vector $U_k v$.

One important property of Φ is that the elements are sorted in a way, that the variance decreases from the first to the last element in each mode [VBPP05]. A reduced model Φ' can be obtained by truncation of insignificant components of Φ and of the matrices U_k, respectively. In the special case of a 2-mode tensor this procedure is equivalent to principal component analysis (PCA) [Vas02].

2.3 Motion Reconstruction

Once we have obtained the reduced model Φ' and its associated matrices U'_k, we are able to reconstruct an approximation of any original motion sequence. This is done by first mode-multiplying the core tensor Φ' with all matrices U'_k, belonging to a technical mode. In a second step the resulting tensor is mode-multiplied with one row of all matrices belonging to a natural mode. Furthermore, with this model at hand, we can generate an arbitrary interpolation of original motions by using linear combinations of rows of the matrices U'_k with respect to the natural modes.

3 Motion Warping

During the last few years, several methods for motion alignment have been proposed which rely on some variant of dynamic time warping (DTW), see, e. g., [BW95, GP00, KG03, MR06]. The alignment or warping result depends on many parameters including the motion features as well as the local cost measure used to compare the features. In this section, we sketch an efficient warping procedure using physics-based motion features (Section 3.1) and applying an iterative multiscale DTW algorithm (Section 3.2).

3.1 Physics-based Features

In our approach, we use physics-based motion features to compare different motion sequences. Physics-based motion features are invariant under global transforms and show a high degree of robustness to spatial variations, which are often present in semantically related motions that belong to the same motion class. Furthermore, our features are still semantically meaningful even on a coarse temporal resolution, which allows us employing them in our multiscale DTW approach.

In our experiments, we used two different types of motion features: the center of mass (COM) acceleration and angular momenta for all skeletal segments. The 3D position of the COM is calculated for all segments of the skeleton by using the anthropometric tables described in [RW03]. From these positions and the mass of the segments one can calculate the COM position of the whole body by summing up the products of the 3D centers of mass of each segment and their corresponding mass and dividing this vector afterwards by the mass of the whole body. The second derivative of the resulting 3D positional data stream is the COM acceleration. Our second feature, the angular momentum, is computed for each segment describing its rotational properties. More precisely the angular momentum how the segments rotation would continue if no external torque acts on it. It is calculated by the cross product between the vector from the point the segment rotates around to the segment's COM and the vector expressing the linear momentum.

Physics-based features provide a lot of information about the underlying motion sequence. For example, considering the COM acceleration it is easy to detect flight phases. More precisely, in case the body has no ground contact, the COM acceleration is equivalent to the acceleration of gravity:

$$a_{\text{COM}} \approx a_{\text{earth}} = \begin{pmatrix} 0.0 \\ -9.81 \\ 0.0 \end{pmatrix}$$

This situation is illustrated by Figure 1, which shows the COM acceleration for a dancing motion. Note that there are three flight phases, which are revealed by the vertical component (y-axis) of the COM acceleration. Further examples are discussed in Section 3.3.

3.2 Multiscale Dynamic Time Warping

Dynamic time warping (DTW) is a well-known technique to find an optimal alignment (encoded by a so-called warping path) between two given sequences. Based on the alignment, the sequences can be warped in a non-linear fashion to match each other. In our context, each motion sequence is converted into a sequence of physics-based motion features at a temporal resolution of 120 Hz. We denote by $V := (v_1, v_2, \ldots, v_n)$ and $W := (w_1, w_2, \ldots, w_m)$ the feature sequences of the two motions to be aligned. Since one of the motions might be slower than the other, n and m do not have to be equal.

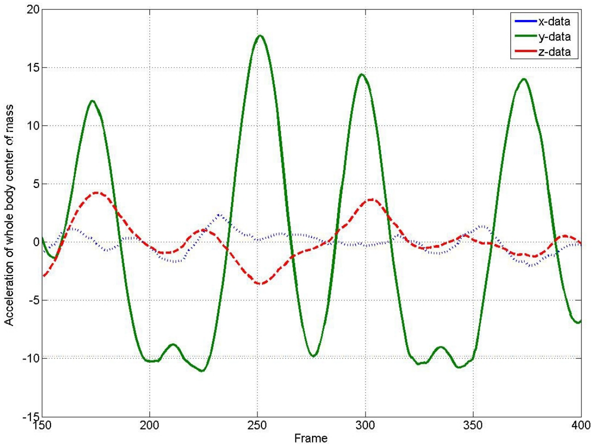

Figure 1: COM acceleration for a dancing motion containing three different jumps. The acceleration is spliced into its x (dotted), y (solid) and z (dashed) component, where the y component refers to the vertical direction. Note that the y component reveals two long flight phases (frames 190 to 220 and frames 320 to 350, respectively) and one short flight phase (around frame 275).

In a second step, one computes an $n \times m$ cost matrix C with respect to some local cost measure c, which is used to compare two feature vectors. In our case, we use a simple cost measure, which is based on the inner product:

$$c(v, w) := 1 - \frac{\langle v|w \rangle}{\|v\|_2 \|w\|_2}$$

for two non-zero feature vectors v and w (otherwise $c(v, w)$ is set to zero). Note that $c(v, w)$ is zero in case v and w coincide and assumes values in the real interval $[0, 1] \subset \mathbb{R}$. Then, the cost matrix C with respect to the sequences V and W is defined by

$$C(i, j) := c(v_i, w_j)$$

for $1 \leq i \leq n$ and $1 \leq j \leq m$. Figure 2 shows such cost matrices with respect to different features.

Finally, an optimal alignment is determined from the cost matrix C via dynamic programming. Such an alignment is represented by a so-called (cost-minimizing) *warping path*, which, under certain constraints, optimally allocates the frame indices of the first motion with the frame indices of the second motion. In Figure 2, such optimal warping paths are indicated in red. Note that the information given by an optimal warping path can be used to time-warp the second motion (by suitably omitting or replicating frames) to match the first motion. Further details and references on DTW may be found in [ZM06].

Note that the time and memory complexity of the DTW algorithm is quadratic in the number of frames of the motions to be aligned. To speed up the process, we employ an iterative multiscale DTW algorithm as described in [ZM06]. Here, the idea is to proceed iteratively using multiple resolution levels going from coarse to fine. In each step, the warping path computed at a coarse resolution level is projected to the next higher level, where the projected path is refined to yield a warping path at the higher level. To obtain features at the coarse levels, we use simple windowing and averaging procedures. In this context, the physics-based features have turned out to yield semantically meaningful features even at a low temporal resolution. In our implementation, we used six different resolution levels starting with a feature resolution of 4 Hz at the lowest level. The overall speed-up of this approach (in comparison to classical DTW) depends on the length of the motion sequences. For example, the speed-up amounts to a factor of roughly 10 for motions having 300 frames and a factor of roughly 100 for motions having 3000 frames.

3.3 Examples

Figure 2 shows two cost matrices, where we compared two walking motions both consisting of 6 steps forward. In dark areas the compared poses are similar with respect to the given features, whereas in lighter areas the poses are dissimilar. The red line is the optimal warping path found by the DTW algorithm. The cost matrix on the left side is based only on the COM acceleration of the entire body. Using this single feature, the checkerboard-like pattern indicates that one cannot differentiate between steps that were done with the left or the right foot. Adding the features that measures the angular momenta of the feet, the result obviously improves. The resulting cost matrix is shown on the right hand side of Figure 2. The five dark diagonals indicate that in this case only the steps made with the same foot are regarded as similar.

Depending on the motions to be time-warped, one can select specific features. For walking motions, the movement of the legs contains the most important information in case the steps are to be synchronized. For time-warping grabbing motions as used in our experiments, aspects concerning the right hand were most important as the motions were performed by this

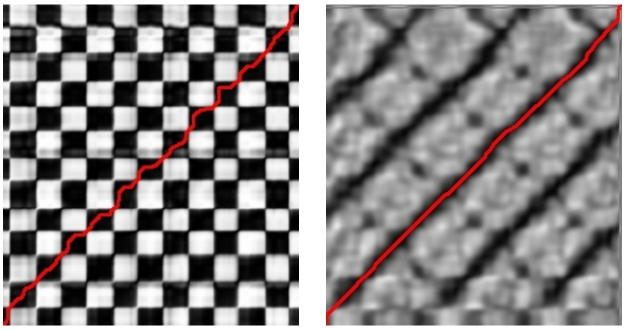

Figure 2: DTW cost matrices calculated on the whole body COM acceleration (left) as well as on the basis of the COM acceleration and the angular momenta of the hands and feet (right). The cost-minimizing warping paths are drawn red.

hand. For our cartwheel motions, good correspondences were achieved when using features that concern the two hands and the two feet. For an example of warped walking motions, we refer to the accompanying video.

4 Experimental Results

4.1 Data Base

For our experiments, we systematically recorded several hours of motion capture data containing a number of well-specified motion sequences, which were executed several times and performed by five different actors. The five actors all have been healthy young adult male persons. Using this data, we built up a database consisting of roughly 210 minutes of motion data. Then we manually cut out suitable motion clips and arranged them into 64 different classes and styles. Each such motion class contains 10 to 50 different realizations of the same type of motion, covering a broad spectrum of semantically meaningful variations. The resulting *motion class database* contains $1,457$ motion clips of a total length corresponding to roughly 50 minutes of motion data [MRC+07]. For our experiments, we considered three motion classes. The first class contains walking motions executed in the following styles:

- Walk four steps in a straight line.
- Walk four steps in a half circle to the left side.
- Walk four steps in a half circle to the right side.
- Walk four steps on the place.

All motions within each of these styles had to start with the right foot and were aligned over time to the length of the first motion of actor one.

The second class of motions we considered in our experiments consists of various grabbing motions, where the actor had to pick an object with the right hand from a storage rack. In this example the style mode corresponds to three different heights(low, middle, and high) the object was located in the rack.

The third motion class consist of various cartwheels. Cartwheel motions were just available for four different actors and for one style. All cartwheels within the class start with the left foot and the left hand.

4.2 Motion Preprocessing

For all motion classes described in the previous section, we constructed data tensors with motion representations based on Euler angles and based on quaternions. Initially some preprocessing was required, consisting mainly of the following steps. All motions were

1. filtered in the quaternion domain with a smoothing filter described as in [LS02],
2. time-warped using physics-based features,
3. normalized by moving the root nodes to the origin and by orienting the root nodes to the same direction,
4. finally sampled down to a frame-rate of 30 Hz.

4.3 Truncation Examples

In this section, we discuss various truncation experiments for our three representative example motion classes. In these experiments, we systematically truncated a growing number of components of the core-tensors, then reconstructed the motions, and compared them with the original motions.

Based on the walking motions (using quaternions to represent the orientations), we constructed two data tensors. The first tensor $\Delta_{\text{Walk}}^{f\cdot n\times a\times s\times r}$ was constructed by using the Data Mode as technical mode. This is indicated by the upper index, which shows the dimension of the tensor. The motions were time-warped and sampled down to a length of 60 frames. The resulting size of $\Delta_{\text{Walk}}^{f\cdot n\times a\times s\times r}$ is $7140 \times 5 \times 4 \times 3$. Using the Frame Mode and DOF Mode, we obtained a second tensor $\Delta_{\text{Walk}}^{f\times n\times a\times s\times r}$ of size $60\times119\times5\times4\times3$. Table 1 shows the results of our truncation experiments. The first column shows the size of the core tensors Φ'_{Walk} after truncation, where the truncated modes are colored red. The second column shows the number of entries of the core tensors, and the third one shows its size in percent compared to Δ_{Walk}. In the fourth column, the total usage of memory is shown. Note that the total memory requirements may be higher than for the original data, since besides the core tensor Φ' one also has to store the matrices U'_k. The memory requirements are particulary high in case one mode has a high dimension. The last two columns give the results of the reconstruction. E_{total} is an error measurement which is defined as the sum over the reconstruction error E_{mot} over all motions:

$$E_{\text{total}} = \frac{1}{a \cdot s \cdot r} \cdot \sum_a \sum_s \sum_r E_{\text{mot}}$$

The reconstruction error E_{mot} of a motion is defined as normalized sum over all frames and over all joints:

$$E_{\text{mot}} = \frac{1}{f \cdot m} \cdot \sum_{i=1}^{f} \sum_{l=1}^{m} (\arccos(\langle q_{i,l}^{\text{org}} | q_{i,l}^{\text{rec}} \rangle) \cdot 2) \cdot \frac{180}{\pi},$$

where f denotes the number of frames and m the number of quaternions. Here, for each joint, the original and reconstructed quaternions q^{org} and q^{rec} are compared by means of their included angle. We performed a visual rating for some of the reconstructed motions in order to obtain an idea of the quality of our error measurement. Here, a reconstructed motion was classified as good (or better) in case one could hardly differentiate it from the original motion when both of the motions were put on top of each other. The results of our ratings are given in the last column.

If the Data Mode is split up into the Frame and DOF Mode, as in $\Delta_{\text{Walk}}^{f \times n \times a \times s \times r}$, one can truncate the two modes separately. The results are shown in the lower part of Table 1 and Figure 3. For example, reducing the DOF Mode from 60 to 26, the error E_{total} is still less than one degree. A similar result is reached by reducing the Frame Mode down to a size of 20. This shows that there is a high redundancy in the data with respect to the technical modes.

We also conducted experiments, where we reduced the dimensions of the natural modes. As the experiments suggest, the dimensions of the natural modes seem to be more important than the ones of the technical modes. The smallest errors (when truncating natural modes) resulted by truncating the Repetition Mode. This is not surprising since the actors were asked to perform the same motion several times in the same fashion. Note that different interpretations by one and the same actor reveal a smaller variance than motions performed by different actors or motions performed in different styles. Some results of our experiments are illustrated by Figure 4. The displacement grows with the size of truncated values from Style- and Personal Mode.

Dimension core tensor	Entries core tensor	Size core tensor in percent	Memory usage in percent	E_{total}	visual rating
Truncation of Data Mode of $\Phi_{\text{Walk}}^{f \cdot n \times a \times s \times r}$					
$7140 \times 5 \times 4 \times 3$	428 400	100 %	12 000 %	0.0000	excellent
$60 \times 5 \times 4 \times 3$	3 600	0.8403 %	100.8520 %	0.0000	excellent
$53 \times 5 \times 4 \times 3$	3 180	0.7423 %	89.0873 %	0.0000	excellent
$52 \times 5 \times 4 \times 3$	3 120	0.7283 %	87.4066 %	0.0634	excellent
$50 \times 5 \times 4 \times 3$	3 000	0.7003 %	84.0453 %	0.2538	very good
$40 \times 5 \times 4 \times 3$	2 400	0.5602 %	67.2386 %	1.1998	very good
$30 \times 5 \times 4 \times 3$	1 800	0.4202 %	50.4318 %	2.1221	very good
$20 \times 5 \times 4 \times 3$	1 200	0.2801 %	33.6251 %	3.6258	good
$10 \times 5 \times 4 \times 3$	600	0.1401 %	16.8184 %	6.3961	good
$5 \times 5 \times 4 \times 3$	300	0.0700 %	8.4150 %	9.3932	satisfying
$4 \times 5 \times 4 \times 3$	240	0.0560 %	6.7344 %	10.4260	satisfying
$3 \times 5 \times 4 \times 3$	180	0.0420 %	5.0537 %	10.9443	sufficient
$2 \times 5 \times 4 \times 3$	120	0.0280 %	3.3730 %	11.5397	poor
$1 \times 5 \times 4 \times 3$	60	0.0140 %	1.6923 %	11.8353	poor
Truncation of Actor Mode of $\Phi_{\text{Walk}}^{f \cdot n \times a \times s \times r}$					
$60 \times 4 \times 4 \times 3$	2 880	0.6723 %	100.6828 %	4.3863	satisfying
$60 \times 3 \times 4 \times 3$	2 160	0.5042 %	100.5135 %	6.4469	satisfying
$60 \times 2 \times 4 \times 3$	1 440	0.3361 %	100.3443 %	8.2369	satisfying
$60 \times 1 \times 4 \times 3$	720	0.1681 %	100.1751 %	10.7773	sufficient
Truncation of Style Mode of $\Phi_{\text{Walk}}^{f \cdot n \times a \times s \times r}$					
$60 \times 5 \times 3 \times 3$	2 700	0.6303 %	100.6410 %	3.5868	good
$60 \times 5 \times 2 \times 3$	1 800	0.4202 %	100.4300 %	5.8414	sufficient
$60 \times 5 \times 1 \times 3$	900	0.2101 %	100.2190 %	8.5770	poor
Truncation of Repetition Mode of $\Phi_{\text{Walk}}^{f \cdot n \times a \times s \times r}$					
$60 \times 5 \times 4 \times 2$	285 600	66,667 %	100.5712 %	2.7639	good
$60 \times 5 \times 4 \times 1$	142 800	33,333 %	100.2904 %	5.0000	good
Truncation of Frame and DOF Mode of $\Phi_{\text{Walk}}^{f \times n \times a \times s \times r}$					
$26 \times 91 \times 5 \times 4 \times 3$	141960	20.9288 %	22.8968 %	0.5492	very good
$21 \times 91 \times 5 \times 4 \times 3$	114660	16.9040 %	18.8020 %	0.7418	very good
$21 \times 46 \times 5 \times 4 \times 3$	57960	8.5449 %	9.6534 %	0.9771	very good
$15 \times 34 \times 5 \times 4 \times 3$	30 600	4.5113 %	5.3252 %	1.9478	good
$14 \times 35 \times 5 \times 4 \times 3$	29 400	4.3344 %	5.1519 %	1.9817	good
$13 \times 37 \times 5 \times 4 \times 3$	28 860	4.2548 %	5.0933 %	1.9724	good
$12 \times 39 \times 5 \times 4 \times 3$	28 800	4.1398 %	4.9994 %	1.9907	good

Table 1: Results for truncating technical and natural modes from our tensors filled with walking motions (using quaternions).

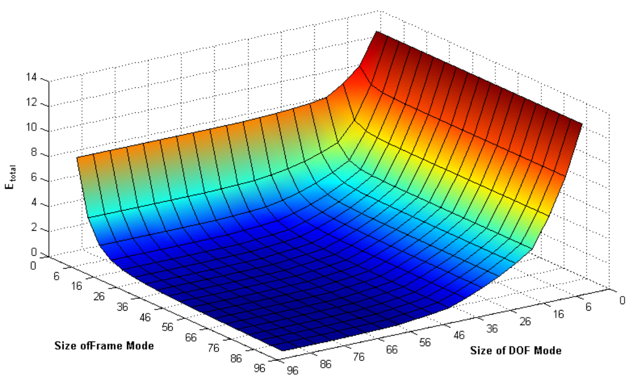

Figure 3: Reconstruction error E_{total} for truncated Frame and DOF Mode.

For building the data tensors $\Delta_{\text{Grab}}^{f \cdot n \times a \times s \times r}$ and

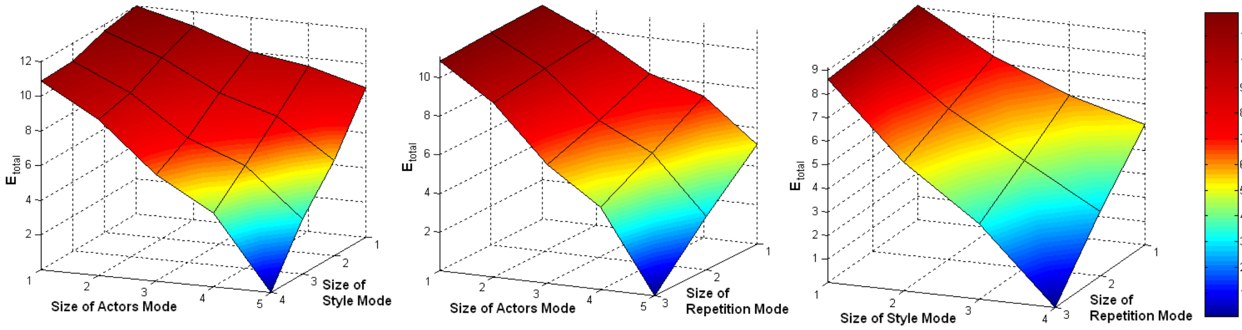

Figure 4: Error E_{total}, of reconstructed motions where two natural Modes were truncated. Actor and Style Mode are truncated (left). Actor and Repetition Mode are truncated (middle). Style and Repetition Mode are truncated (right).

Dimension core tensor	Entries core tensor	Size core tensor in percent	Memory usage in percent	E_{total}	visual rating
Truncation of Data Mode of $\Phi^{f\cdot n\times a\times s\times r}_{\text{Grab}}$					
$8449 \times 5 \times 3 \times 3$	380 205	100 %	18 775 %	0.0000	excellent
$60 \times 5 \times 3 \times 3$	2 700	0.7101 %	134.0548 %	0.0000	excellent
$55 \times 5 \times 3 \times 3$	2 475	0.6510 %	122.8845 %	0.0000	excellent
$50 \times 5 \times 3 \times 3$	2 250	0.5918 %	111.7142 %	0.0000	excellent
$45 \times 5 \times 3 \times 3$	2 025	0.5326 %	100.5439 %	0.0000	excellent
$40 \times 5 \times 3 \times 3$	1 800	0.4734 %	89.3736 %	1.2632	very good
$35 \times 5 \times 3 \times 3$	1 575	0.4143 %	78.2033 %	2.1265	very good
$30 \times 5 \times 3 \times 3$	1 350	0.3551 %	67.0330 %	2.9843	very good
$25 \times 5 \times 3 \times 3$	1 125	0.2959 %	55.8628 %	3.9548	good
$20 \times 5 \times 3 \times 3$	900	0.2367 %	44.6925 %	5.1628	good
$15 \times 5 \times 3 \times 3$	675	0.1775 %	33.5222 %	6.6799	satisfying
$10 \times 5 \times 3 \times 3$	450	0.1184 %	22.3519 %	8.8702	sufficient
$5 \times 5 \times 3 \times 3$	225	0.0592 %	11.1816 %	11.5604	sufficient
$4 \times 5 \times 3 \times 3$	180	0.0473 %	8.9475 %	12.4463	poor
$3 \times 5 \times 3 \times 3$	135	0.0355 %	6.7135 %	12.7304	poor
$2 \times 5 \times 3 \times 3$	90	0.0237 %	4.4794 %	13.4234	poor
$1 \times 5 \times 3 \times 3$	45	0.0118 %	2.2454 %	13.7150	poor
Truncation of DOF Mode of $\Phi^{f\times n\times a\times s\times r}_{\text{Grab}}$					
$71 \times 91 \times 5 \times 3 \times 3$	290 745	76.4706 %	80.6560 %	0.0000	excellent
$71 \times 86 \times 5 \times 3 \times 3$	274 770	72.2689 %	76.2978 %	0.0001	excellent
$71 \times 61 \times 5 \times 3 \times 3$	194 895	51.2605 %	54.5069 %	0.1311	excellent
$71 \times 51 \times 5 \times 3 \times 3$	162 945	42.8571 %	45.7906 %	0.4332	good
$71 \times 41 \times 5 \times 3 \times 3$	130 995	34.4538 %	37.0742 %	1.0450	satisfying
$71 \times 31 \times 5 \times 3 \times 3$	99 045	26.0504 %	28.3579 %	2.2182	sufficient
$71 \times 21 \times 5 \times 3 \times 3$	67 095	17.6471 %	19.6415 %	3.9491	sufficient
$71 \times 11 \times 5 \times 3 \times 3$	35 145	9.2437 %	10.9252 %	7.1531	poor
$71 \times 6 \times 5 \times 3 \times 3$	19 170	5.0420 %	6.5670 %	10.1546	poor
$71 \times 1 \times 5 \times 3 \times 3$	3 195	0.8403 %	2.2088 %	14.3765	poor

Table 2: Truncation results for grabbing motions (using quaternions).

$\Delta^{f\times n\times a\times s\times r}_{\text{Grab}}$, all motions were warped to the length of one reference motion. $\Delta^{f\cdot n\times a\times s\times r}_{\text{Grab}}$ has a size of $8449 \times 5 \times 3 \times 3$, while $\Delta^{f\times n\times a\times s\times r}_{\text{Grab}}$ has a size of $71 \times 119 \times 5 \times 3 \times 3$. The exact values for truncating the Data Mode and the DOF Mode can be found in Table 2.

In our third example, we consider a motion class consisting of cartwheel motions. The core tensor $\Delta^{f\cdot n\times a\times s\times r}_{\text{Cart}}$ has a size of $7497 \times 4 \times 1 \times 3$. Here, all motions could be reconstructed without any visible error for a size of no more than 12 dimensions for the Data Mode. Further results are shown in Table 3.

The number of necessary components of the data mode varied a lot in the different motion classes. One would expect that a cartwheel motion is more complex than a grabbing or walking motion. The results of previous experiments do not support this prospect. But the results of this truncation experiments are not comparable as they depend on all dimensions of the constructed tensors. To get comparable results for the three motion classes, we constructed a tensor including three motions from three actors for each motion class. The style mode is limited to a size of one, since we have no different styles for cartwheel motions. Therefor the resulting tenors have a size of $f \cdot n \times 3 \times 3 \times 1$. When truncating the data mode of these tensors, one gets the result that is shown in Figure 5. All motions are reconstructed perfectly until the size of the data mode gets smaller than 12. At this size the core tensor Φ' and the matrices U_k have as many entries as the original data tensor Δ. Then the error E_{total} grows different for the three motion classes. The smallest error is observed for the walking motions (solid). This could be expected for a cyclic motion that contains a lot of redundant frames. The used grabbing motions (dotted) are more complex. The reason may be that the motions sequences disparate since some sequences include a step to the storage rack while others

Dimension core tensor	Entries core tensor	Size core tensor in percent	Memory usage in percent	E_{total}	visual rating
Truncation of Data Mode of $\Phi^{f\cdot n\times a\times s\times r}_{\text{Cart}}$					
$30 \times 4 \times 3$	360	0.4002 %	250.4279 %	0.0000	excellent
$12 \times 4 \times 3$	144	0.1601 %	100.1879 %	0.0000	excellent
$11 \times 4 \times 3$	132	0.1467 %	91.8412 %	1.6780	good
$10 \times 4 \times 3$	120	0.1334 %	83.4945 %	3.2163	satisfying
$9 \times 4 \times 3$	108	0.1200 %	75.1478 %	5.7641	sufficient
$8 \times 4 \times 3$	96	0.1067 %	66.8012 %	8.1549	poor
$5 \times 4 \times 3$	60	0.0667 %	41.7611 %	13.2847	poor
$1 \times 4 \times 3$	12	0.0133 %	8.3745 %	22.4095	poor

Table 3: Truncation results for cartwheel motions (using quaternions).

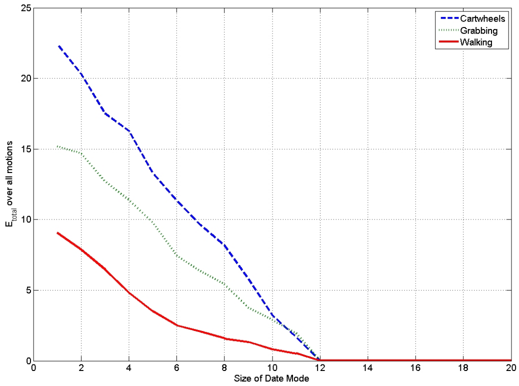

Figure 5: Reconstruction error E_{total} for walking (solid), grabbing (dotted) and cartwheel (dashed) motions, depending on the size of the Data Mode of the core tensor.

do not. The cartwheel motions (dashed) are the most complex class in this experiment, as we expected.

4.4 Comparison with PCA

To compare our multilinear model with linear models, as they are used for principal component analysis (PCA), we constructed two tensors for our model and two matrices for the PCA. The first tensor and the first matrix were filled with walking motions, the second tensor and the second matrix were filled with grabbing motions. The orientations were represented by Euler angles. The resulting tensors had a size of $81 \times 62 \times 3 \times 3 \times 3$ (walking) and $64 \times 62 \times 3 \times 3 \times 3$ (grabbing), respectively. After some data reduction step, we compared the reconstructed motions with the original motions by measuring the differences between all orientations of the original and the reconstructed motions. Averaging over all motions and differences, we obtained a mean error as is also used in [SHP04] (we used this measure to keep the results comparable to the literature). Figure 6 shows a comparison of the mean errors in the reconstructed motions for the walking (left) and grabbing (right) examples. The mean errors depend on the size of the DOF Mode and the number of principal components, respectively. Note that the errors for motions reconstructed from the multi-mode-model are smaller than the errors from the motions reconstructed from principal components. For example, a walking motion can be reconstructed with a mean error of less than one degree (in the average) from a core tensor when the DOF Mode is truncated to just three components (see left part of Figure 6). Therefore, in cases where a motion should be approximated by rather few components the reduction based on the multilinear model may be considerably better than the one achieved by PCA.

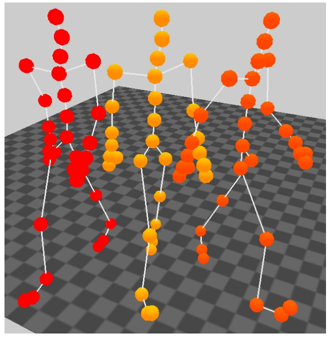

Figure 7: Screenshot from the original motions that are from the styles walking forward (left) and walking a left circle (right), the synthetic motion (middle) is produced by a linear combination of these styles.

4.5 Motion Synthesis

As it was described in Sect. 2.3, it is possible to synthesize motions with our multilinear model. For every mode k there is an appropriate matrix U_k, where every row $u_{k,j}$ with $j \in \{1, 2, \ldots, d_k\}$ represents one of the dimensions, of mode k. Therefore an inter- or extrapolation between the d_k dimensions e.g. between the styles, actors and repetitions, can be done by inter- or extrapolation between any rows of U_k before they are multiplied with the core tensor Φ to synthesize a motion. To prevent our results from artifacts such as turns and unexpected flips resulting from a representation based on Euler angles we used our quaternion based representation to synthesize motions.

For the following walking example, we constructed a motion that was interpolated between two different styles. The first style was walking four steps straight forward and the second one was walking four steps on a left circle. We made a linear interpolation by multiplying the corresponding rows with the factor 0.5. The result is a four step walking motion that describes a left round with a larger radius. One sample frame of this experiment can be seen in Figure 7. Another synthetic motion was made by an interpolation of grabbing styles. We synthesized a motion by an interpolation of the styles grabbing low and grabbing high. The result is a motion that grabs in the middle. One sample frame of this synthetic motion is shown in Figure 8.

With this technique we are able to make interpolation between all modes simultaneous. One example

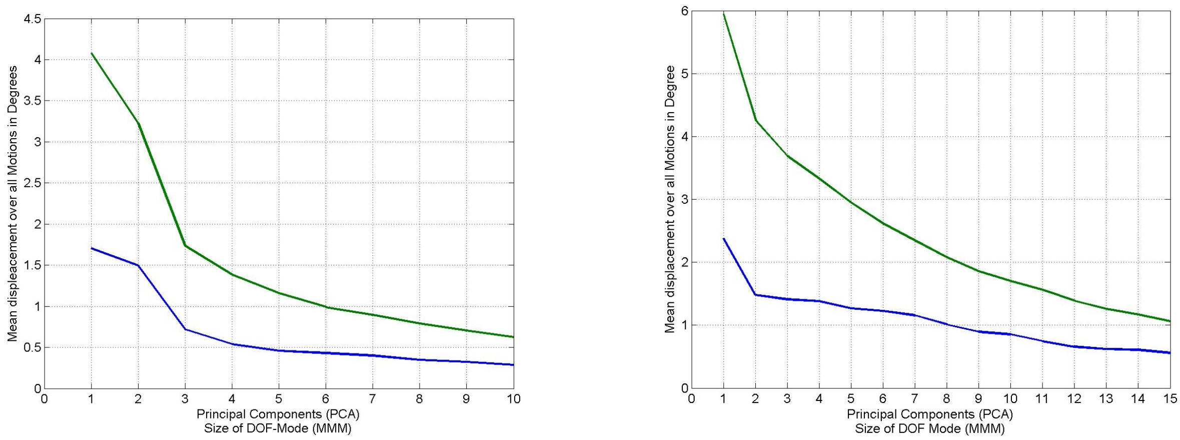

Figure 6: Mean error of reconstructed motions with reconstructions based on our model (blue) and based on a PCA (green). The result is shown for walking motions (left) and grabbing motions (right).

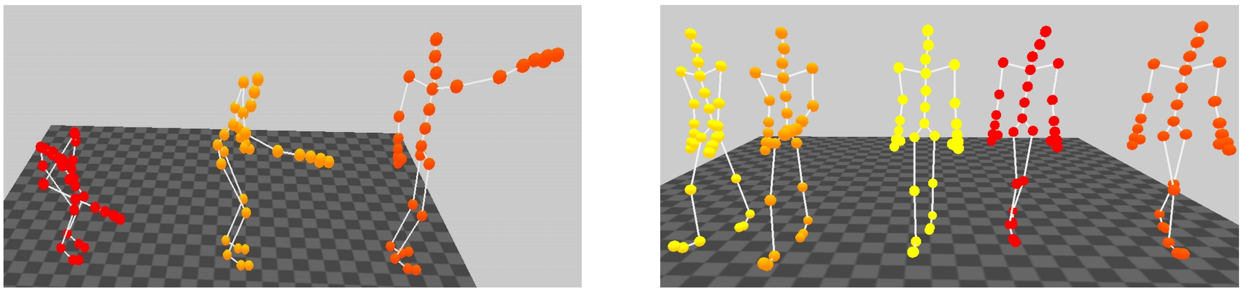

Figure 8: **Left:** Screenshot from the original motions that are from the styles grabbing low (left) and grabbing high (right), the synthetic motion (middle) is produced by a linear combination of these styles. **Right:** Screenshot from four original walking motions and one synthetic motion, that is an result of combining two, the personal and the style mode. The original motions of the first actor are on the left side, the original motions of the second actor are on the right side and the synthetic example can be seen in the middle.

is a walking motion that is an interpolation between the Style and Actors Mode. One snapshot taken from the accompanying animation video of this example is given in Figure 8.

4.6 Computation Times

In Table 4 the computation times of our MATLAB implementations of the N-Mode SVD and PCA are given (on an 1.66 GHz Intel Core2 CPU T5500). For the decomposition of a data tensor $\Delta_{\text{Walk}}^{f\times n\times a\times s\times r}$ consisting of $95 \times 119 \times 4 \times 3 \times 2 = 273600$ entries, the N-Mode SVD needs 5.675 seconds, while the PCA needs 0.636 seconds for a matrix of comparable size having $119 \times 2280 = 273600$ entries. As Table 4 shows, the computing time for the SVD increase with the dimension of the tensor, while the computation time for the PCA is nearly constant.

The SVD decomposition can be seen as a preprocessing step, where all further calculations can be done on the core tensor and the corresponding matrices. The reconstruction of a motion from the tensor $\Phi_{\text{Walk}}^{f\times n\times a\times s\times r}$ and the matrices U_k can be performed at interactive frame rates—even in our MATLAB implementation the reconstruction only requires 0.172 seconds. As a combination of motions of different modes is just a reconstruction with modified weights, the creation of synthetic motions is also possible with a similar computational cost.

Dimension Motion Matrix	**Time PCA (in sec.)**	**Dimension Core Tensor**	**Time N-mode SVD (in sec.)**
119×2280	0.636	$95 \times 119 \times 4 \times 3 \times 2$	5.675
95×2280	0.621	$95 \times 95 \times 4 \times 3 \times 2$	4.968
80×2280	0.615	$95 \times 80 \times 4 \times 3 \times 2$	4.461
5×2280	0.600	$95 \times 5 \times 4 \times 3 \times 2$	2.295

Table 4: Computation times for PCA and N-Mode-SVD for the data used in the examples.

5 Conclusion and Future Work

In this paper, we have shown how multilinear models can be used for analyzing and processing human motion data. The representation is based on explicitly using various modes that correspond to technical as well as semantic aspects of some given motion class. Encoding the data as high-order tensors allows for reducing the model with respect to any combination of modes, which often yields better approximation results than previously used PCA-based methods. Furthermore, the multilinear model constitutes a unified and flexible framework for motion synthesis applications, which allows for controlling each motion aspect independently in the morphing process. As a further contribution, we described an efficient multiscale approach for motion warping using physics-based motion features.

Multilinear motion representations constitute an interesting alternative and additional tool in basically all situations, where current PCA-based methods are used. We expect that our multi-modal model is helpful in the context of reconstructing motions from low-dimensional control signals, see, e. g., [CH05]. Currently, we also investigate how one can improve auditory representations of motions as described in [RM05, EMWZ05] by using strongly reduced motion representations.

In order to construct a high-order tensor for a given motion class, one needs a sufficient number of example motions for each mode to be considered in the model. In practice, this is often problematic, since one may only have sparsely given data for the different modes. In such situations, one may employ similar techniques as have been employed in the context of face transfer, see [VBPP05], to fill up the data tensor. Another important research problem concerns the automatic extraction of suitable example motions from a large database, which consists of unknown and unorganized motion material. For the future, we plan to employ efficient content-based motion retrieval strategies as described, e. g., in [KG04, MRC05, MR06] to support the automatic generation of multimodal data tensors for motion classes that have a sufficient number of instances in the unstructured dataset.

Acknowledgements

We are grateful to Bernd Eberhardt and his group from HDM Stuttgart for providing us with the mocap data. We also thank the anonymous referees for constructive and valuable comments.

Citation
Björn Krüger, Jochen Tautges, Meinard Müller, and Andreas Weber, *Multi-Mode Tensor Representation of Motion Data*, Journal of Virtual Reality and Broadcasting, 5(2008),no. 5, July 2008, urn:nbn:de:0009-6-14197, ISSN 1860-2037.

References

[BA00] Rasmus Bro and Claus A. Anderssono, *The N-way Toolbox for MATLAB*, 2000, http://www.models.kvl.dk/source/-nwaytoolbox/.

[BSP+04] Jernej Barbič, Alla Safonova, Jia-Yu Pan, Christos Faloutsos, Jessica K. Hodgins, and Nancy S. Pollard, *Segmenting motion capture data into distinct behaviors*, Proceedings of the Graphics Interface 2004 Conference (Wolfgang Heidrich and Ravin Balakrishnan, eds.), Canadian Human-Computer Communications Society, 2004, ISBN 1-56881-227-2, pp. 185–194.

[BW95] Armin Bruderlin and Lance Williams, *Motion signal processing*, Proceedings of the 22nd annual conference on Computer graphics and interactive techniques (Robert Cook, ed.), ACM Press, New York, 1995, ISBN 0-80791-701-4, pp. 97–104.

[CH05] Jinxiang Chai and Jessica K. Hodgins, *Performance animation from low-dimensional control signals*, ACM Trans. Graph. **24** (2005), no. 3, 686–696, ISSN 0730-0301.

[CH07] Jinxiang Chai and Jessica K. Hodgins, *Constraint-based Motion Optimization Using A Statistical Dynamic Model*, ACM Transactions on Graphics **26** (2007), no. 3, SIGGRAPH 2007, ISSN 0730-0301.

[EMWZ05] Alfred Effenberg, Joachim Melzer, Andreas Weber, and Arno Zinke, *MotionLab Sonify: A Framework for the Sonification of Human Motion Data*, Ninth International Conference on Information Visualisation (IV'05) (London, UK), IEEE Press, July 2005, ISBN 0-7695-2397-8, pp. 17–23.

[FF05] Kate Forbes and Eugene Fiume, *An efficient search algorithm for motion data using weighted PCA*, Proc. 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2005, ISBN 1-7695-2270-X, pp. 67–76.

[GBT04] Pascal Glardon, Ronan Boulic, and Daniel Thalmann, *PCA-Based Walking Engine Using Motion Capture Data*, CGI '04: Proceedings of the Computer Graphics International (CGI'04) (Washington, DC, USA), IEEE Computer Society, 2004, ISBN 0-7695-2171-1, pp. 292–298.

[GP00] Martin A. Giese and Tomaso Poggio, *Morphable Models for the Analysis and Synthesis of Complex Motion Patterns*, International Journal of Computer Vision **38** (2000), no. 1, 59–73, ISSN 0920-5691.

[HPP05] Eugene Hsu, Kari Pulli, and Jovan Popović, *Style translation for human motion*, ACM Trans. Graph. **24** (2005), no. 3, 1082–1089, ISSN 0730-0301.

[KG03] Lucas Kovar and Michael Gleicher, *Flexible Automatic Motion Blending with Registration Curves*, Eurographics/SIGGRAPH Symposium on Computer Animation (D. Breen and M. Lin, eds.), Eurographics Association, 2003, ISBN 1-58113-659-5, pp. 214–224.

[KG04] Lucas Kovar and Michael Gleicher, *Automated extraction and parameterization of motions in large data sets*, ACM Transactions on Graphics **23** (2004), no. 3, 559–568, SIGGRAPH 2004, ISSN 0730-0301.

[KGP02] Lucas Kovar, Michael Gleicher, and Frédéric Pighin, *Motion Graphs*, ACM Transactions on Graphics **21** (2002), no. 3, 473–482, SIGGRAPH 2002, ISSN 0730-0301.

[LS02] Jehee Lee and Sung Yong Shin, *General Construction of Time-Domain Filters for Orientation Data*, IEEE Transactions on Visualizatiuon and Computer Graphics **8** (2002), no. 2, 119–128, ISSN 1077-2626.

[LZWM05] Guodong Liu, Jingdan Zhang, Wei Wang, and Leonard McMillan, *A system for analyzing and indexing human-motion databases*, Proc. 2005 ACM SIGMOD Intl. Conf. on Management of Data, ACM Press, 2005, ISBN 1-59593-060-4, pp. 924–926.

[MK06] Tomohiko Mukai and Shigeru Kuriyama, *Multilinear Motion Synthesis Using Geostatistics*, ACM SIGGRAPH / Eurographics Symposium on Computer Animation - Posters and Demos, 2006, pp. 21–22.

[MR06] Meinard Müller and Tido Röder, *Motion Templates for Automatic Classification and Retrieval of Motion Capture Data*, SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, ACM Press, 2006, ISBN 3-905673-34-7, pp. 137–146.

[MRC05] Meinard Müller, Tido Röder, and Michael Clausen, *Efficient content-based retrieval of motion capture data*, ACM Trans. Graph. **24** (2005), no. 3, 677–685, ISSN 0730-0301.

[MRC+07] Meinard Müller, Tido Röder, Michael Clausen, Bernhard Eberhardt, Björn Krüger, and Andreas Weber, *Documentation: Mocap Database HDM05*, Computer Graphics Technical Report CG-2007-2, Universität Bonn, June 2007, ISSN 1610-8892.

[MZF06] Anna Majkowska, Victor B. Zordan, and Petros Faloutsos, *Automatic Splicing for Hand and Body Animations*, ACM SIGGRAPH / Eurographics Symposium on Computer Animation, 2006, ISBN 3-905673-34-7, pp. 309–316.

[OBHK05] Dirk Ormoneit, Michael J. Black, Trevor Hastie, and Hedvig Kjellström, *Representing cyclic human motion using functional analysis*, Image Vision Comput. **23** (2005), no. 14, 1264–1276, ISSN 0262-8856.

[RCO05] Kalanov Rovshan, Jieun Cho, and Jun Ohya, *D-12-79 A Study of Synthesizing New Human Motions from Sampled Motions Using Tensor Decomposition*, Proceedings of the IEICE General Conference **2005** (2005), no. 2, 229.

[RM05] Niklas Röber and Maic Masuch, *Playing Audio-only Games: A compendium of interacting with virtual, auditory Worlds*, Proceedings of Digital Games Research Conference (Vancouver, Canada), 2005.

[RW03] K. L. Robbins and Q. Wu, *Development of a Computer Tool for Anthropometric Analyses*, Proceedings of the International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences (METMBS'03) (Las Vegas, USA) (Faramarz Valafar and Homayoun Valafar, eds.), CSREA Press, June 2003, ISBN 1-932415-04-1, pp. 347–353.

[SH05] Alla Safonova and Jessica K. Hodgins, *Analyzing the physical correctness of interpolated human motion*, SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation (New York, NY, USA) (Demetri Terzopoulos ans Victor Zordan, ed.), ACM Press, 2005, ISBN 1-59593-198-8, pp. 171–180.

[SH07] Alla Safonova and Jessica K. Hodgins, *Construction and Optimal Search of Interpolated Motion Graphs*, ACM Transactions on Graphics **26** (2007), no. 3, SIGGRAPH 2007, ISSN 0730-0301.

[SHP04] Alla Safonova, Jessica K. Hodgins, and Nancy S. Pollard, *Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces*, ACM Transactions on Graphics **23** (2004), no. 3, 514–521, ISSN 0730-0301.

[Tro02] Nikolaus F. Troje, *Decomposing biological motion: A framework for analysis and synthesis of human gait patterns*, Journal of Vision **2** (2002), no. 5, 371–387, ISSN 1534-7362.

[Vas02] M. Alex O. Vasilescu, *Human motion signatures: Analysis, synthesis, recognition*, Proc. Int. Conf. on Pattern Recognition (Quebec City, Canada), 2002, ISBN 0-7695-1695-X, pp. 456–460.

[VBPP05] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović, *Face transfer with multilinear models*, ACM Trans. Graph. **24** (2005), no. 3, 426–433, SIGGRAPH 2005, ISSN 0730-0301.

[ZM06] Arno Zinke and Dessislava Mayer, *Iterative Multi Scale Dynamic Time Warping*, Technical Report CG-2006-1, Universität Bonn, 2006, ISSN 1610-8892.